LAMP-TR-081 February 2002
CS-TR-4332
UMIACS-TR-2002-14

Generating A Parsing Lexicon From Lexical-Conceptual Structure

Necip Fazil Ayan, Bonnie J. Dorr

Language and Media Processing Labratory
Instititue for Advanced Computer Studies
College Park, MD 20742

Abstract

This paper describes the generation of a lexicon for a principle-based parser (Minipar [5,6]) using descriptions of verbs based on Lexical-Conceptual Structure (LCS [1,2]).1 We aim to determine how much syntactic information we can obtain from a semantic-based lexicon. More specifically, we aim to provide a general approach to projection of syntactic entries from semantic (language-independent) lexicons–and to test the effect of such lexicons on parser performance. Verbs are grouped together into classes–each denoted by an LCS representation adn the thematic grid. These are mapped systematically into syntactic categories associated with entries in the Minipar parser. The main advantage of this LCS-to-syntax projection is language protability: We currently have LCS lexicons for English, Arabic, Spanish, and Chinese; thus, our LCS-projection approach allows us to produce syntactic lexicons for parsing in each of these languages. For comparing the performance of the projection from the LCS to Minpar coes, we also generated the mappings for the codes of Longman's Directionary of Contemporary English (LDOCE [8])–the most comprehensive online dictionary for syntactic categorization. Preliminary experiments indicate that our approach yields a categorization of verbs with 58% precision and 65% recall as measured against LDOCE–with an improved precision of 74% when redundancy is removed. The next section presents a brief description of each code set we use. In Section 3, we explain how we generated Minipar codes from LCS representation. Finally, Section 4, discusses our experiments and results.

***The support of the LAMP Technical Report Series and the partial support of this research by the National Science Foundation under grant EIA0130422 and the Department of Defense under contract MDA9049-C6-1250 is gratefully acknowledged.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
FEB 2002		**00-02-2002 to 00-02-2002**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Generating a Parsing Lexicon From Lexical-Conceptual Structure	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Language and Media Processing Laboratory,Institute for Advanced Computer Studies,University of Maryland,College Park,MD,20742-3275**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**5**	

Generating A Parsing Lexicon From Lexical-Conceptual Structure

Necip Fazil Ayan and Bonnie J. Dorr

February 14, 2002

Category: Research Paper.

1 Introduction

This paper describes the generation of a lexicon for a principle-based parser (Minipar [5, 6]) using descriptions of verbs based on Lexical-Conceptual Structure (LCS [1, 2]).[1] We aim to determine how much syntactic information we can obtain from a semantic-based lexicon. More specifically, we aim to provide a general approach to projection of syntactic entries from semantic (language-independent) lexicons—and to test the effect of such lexicons on parser performance.

Verbs are grouped together into classes—each denoted by an LCS representation and a thematic grid. These are mapped systematically into syntactic categories associated with entries in the Minipar parser. The main advantage of this LCS-to-syntax projection is language portability: We currently have LCS lexicons for English, Arabic, Spanish, and Chinese; thus, our LCS-projection approach allows us to produce syntactic lexicons for parsing in each of these languages.

For comparing the performance of the projection from LCS to Minipar codes, we also generated the mappings for the codes of Longman's Dictionary of Contemporary English (LDOCE [8])—the most comprehensive online dictionary for syntactic categorization. Preliminary experiments indicate that our approach yields a categorization of verbs with 58% precision and 65% recall as measured against LDOCE—with an improved precision of 74% when redundancy is removed.

The next section presents a brief description of each code set we use. In Section 3, we explain how we generated Minipar codes from the LCS representation. Finally, Section 4, discusses our experiments and results.

2 Code Descriptions

In many online dictionaries, verbs are classified according to the arguments and modifiers that they allow. Most dictionaries use specific codes to identify transitivity, intransitivity, and ditransitivity. These broad categories may be further refined, e.g., to distinguish verbs with NP arguments from those with clausal arguments. The degree of refinement varies widely. We examine two schemes: Minipar and LDOCE.

2.1 Minipar Codes

The Minipar coding scheme is an adaptation of the codes used in the Oxford Advanced Learner's Dictionary (OALD [7]). Verbs are categorized into 5 main groups: Intransitive, transitive, ditransitive, complex transitive, and linking. Each code is of the form $Sa_1[.a_2]$ where S is the first letter of the verb categorization ($S \in \{L, I, T, C, D\}$ for the corresponding groups), and $a_1, a_2, \ldots$ are the argument types. If a code contains more than 1 argument, each argument is listed serially. Possible argument types are n for nouns, f for finite clauses ("that" clauses), g for -ing clauses, t for infinitive clauses, w for finite clauses beginning with -wh questions, i for bare infinitive clauses, a for adjectives, p for prepositions and pr for PPs. The number of codes in OALD is 32. (See Table 1.)

[1] We focus only on verb entries as they are cross-linguistically the most complex constituents.

Categorization	OALD Codes
Intransitive verbs	{I, Ip, Ipr, In/pr, It}
Transitive verbs	{Tn, Tn.pr, Tn.p, Tf, Tw, Tt, Tg, Tn.t, Tn.g, Tn.i}
Complex Transitive verbs	{Cn.a, Cn.n, Cn.n/a, Cn.t, Cn.g, Cn.i}
Ditransitive verbs	{Dn.n, Dn.pr, Dn.f, Dn.t, Dn.w, Dpr.f, Dpr.w, Dpr.t}
Linking verbs	{La, Ln}

Table 1: OALD Code Set: The Basis of Minipar Codes

Number	Arguments
1	one or more nouns
2	bare infinitive clause
3	infinitive clause
4	-ing form
5	-that clause
6	clauses with a wh- word
7	adjective
8	past participle
9	descriptive word or phrase

Table 2: LDOCE Number Description

OALD codes are simplistic in that they do not include modifiers. In addition, they also do not explicitly specify which prepositions can be used in the PPs.[2] Minipar extends the OALD codes by providing a facility for specifying prepositions, but only 8 verbs are encoded with these prepositional codes in the official Minipar distribution. In these cases, the codes containing *pr* are refined to be *pr.prep*, where *prep* is the head of the PP argument.[3] In addition, Minipar codes are refined in the following ways:

1. Optional arguments are allowed, e.g., T[n].pr describes verbs followed by an optional noun and a PP.
2. Two or more codes may be combined, e.g., Tfgt describes verbs followed by a finite, gerund, or infinitive clause.

There are 66 Minipar codes.

2.2 LDOCE Codes

LDOCE has a more detailed code set than that of OALD (and hence Minipar). The codes include both arguments and modifiers. Moreover, prepositions are richly specified throughout the lexicon. The syntax of the codes is either *CN* or *CN-Prep*, where *C* is the initial letter of the verb sub-categorization (as in the generic OALD codes) and N is a number, which corresponds to different sets of arguments that can follow the verb. For example, T1-ON refers to verbs that are followed by a noun and a PP with the head "on". The number of codes included in this set is 179. The meaning of each is described in Table 2.

3 Our Approach

The LCS lexicon consists of verbs grouped into classes based on [4] along with the thematic grid structures of the verbs (see [2]). Our new parsing lexicon uses an enhanced OALD-style encoding (henceforth referred to as OALD+) that accommodates modifiers, optionality of arguments and modifiers, a more rich set of prepositions, and multiple PP arguments.

To generate this new parsing lexicon, we performed the following steps:

[2] For example, *Tn* refers to a verb followed by a noun ('She read the book'), *Tn.pr* refers to a verb followed by a noun and a PP ('He opened the door with a latch'), and *Dn.n* refers to a verb followed by two nouns ('She taught the children French').

[3] This extension is used only for the preposition *as* for the verbs *absolve, accept, acclaim, brand, designate, disguise, fancy,* and *reckon*.

	Minipar Lexicon	LCS Lexicon	Intersection
Precision	81.2%	57.9%	92.1%
Recall	62.3%	64.4%	50.4%

Table 3: Precision and Recall of Code Sets

1. Map the entries in the LCS lexicon to OALD+
2. Produce a pairing of verbs to OALD+ codes from step 1

For comparing effectiveness of our projection of OALD+ entries from LCS entries, we examined the degree lexical coverage of our enhanced lexicon and that of the original Minipar lexicon, using LDOCE as our "gold standard". More precisely, we compared the precision and recall our OALD+ entries with the precision and recall of the original Minipar lexicon.

In order to retain equivalence, we converted LDOCE entries into the analogous OALD+ format. Thus, we implemented three mappings, detailed below.

3.1 LCS-to-OALD

We automatically assigned OALD+ code(s) for each verb in the LCS lexicon using its semantic class number and thematic grid. Our OALD+ specifies prepositions for entries that require them. For example, the grid _ag_th_instr(with) is mapped into the code Tn.pr.with instead of a generic OALD code of Tn.pr. In addition, OALD+ codes accommodates optional arguments, modifiers, and constructions containing more than one PP argument. For example, Tn.pr.pr is an example of transitive verb with two PPs ("She drove the kids from home to school").

3.2 Minipar-to-OALD+

We generated OALD+ codes from Minipar codes simply using the reverse mapping described in Section 2.1.

3.3 LDOCE-to-OALD+

Each LDOCE code was mapped manually to one or more OALD+ codes. LDOCE codes are more refined than the generic OALD codes, but the OALD+ codes accommodate these refinements. For example, D1-AT and T1-AT are mapped into Dn.pr.at in the OALD code set.

4 Results

In our comparison, we considered the three lexicons resulting from the mappings above: The first contains OALD+ codes based on LCS; the second contains OALD+ codes based on Minipar; and the third contains OALD+ codes based on LDOCE—this last one serves as our "gold standard". Our precision and recall results thus are based on the following inputs:

A = Number of OALD+ verb/code pairs in L occurring in LDOCE
B = Number of OALD+ verb/code pairs in L NOT occurring in LDOCE
C = Number of OALD+ verb/code pairs in LDOCE NOT occurring in L

Given an OALD+ encoded lexicon L, we compute: (1) The *precision* of L = $A/(A+B)$; and (2) the *recall* of L = $A/(A+C)$. We compare two results: one where L is the Minipar-based lexicon and one where L is the LCS-based lexicon.

Given that the number of the verbs in each lexicon is different and neither of them completely covers one another, we take only those verbs that occur in both L and LDOCE, for each L. We also conducted an experiment where we take the intersection of OALD+ verb/code pairs that occur both in LCS lexicon and Minipar lexicon. Table 3 displays the results. The highest precision is achieved by the intersection of two lexicons, but at the expense of

recall. We found that the precision was higher for Minipar than for the LCS lexicon, but when we examined this in more detail, we found that this was almost entirely due to "double counting" of entries with optional modifiers in the LCS-based lexicon. For example, the single LCS-based grid _ag_th,instr(with) corresponds to two OALD+ codes, Tn and Tn.pr while LDOCE takes this only as Tn. Specifically, 43% of the redundant codes in LCS lexicon (the ones that are in LCS lexicon but not in LDOCE) are Tn.pr and 96% of them occur with also Tn for the same verb. Similarly, 16% of the redundant codes are Ipr, with 77% of them together with I.

Neither LDOCE nor Minipar take modifiers into account; thus, the LCS-based lexicon is spuriously low in its precision. To observe the degree of the impact of optional modifiers, we computed another precision value for LCS lexicon as follows: While computing precision, if the redundant code for a verb includes a PP (i.e. ".pr") and the verb also contains a code without the PP, we do not count it twice. With this methodology, we achieved 74% precision.

We conclude that it is possible to produce a parsing lexicon by projecting from LCS-based lexical entries—achieving precision and recall on a par with a syntactic lexicon (Minipar) encoded by hand specifically for English. The consequence of this result is that, as semantic lexicons become increasingly available for multiple languages (ours are now available in English, Chinese, and Arabic), we are able to produce parsing lexicons automatically for each language. Our future work involves the integration of these parsing lexicons in ongoing machine translation work [3].

Acknowledgements

This work has been supported, in part, by ONR MURI Contract FCPO.810548265, DARPA TIDES Contract N66001-00-2-8910, and DOD Contract MDA904-96-C-1250.

References

[1] Bonnie J. Dorr. *Machine Translation: A View from the Lexicon.* The MIT Press, Cambridge, MA, 1993.

[2] Bonnie J. Dorr. LCS Verb Database. Technical Report Online Software Database, University of Maryland, College Park, MD, 2001. http://www.umiacs.umd.edu/~bonnie/LCS_Database_Docmentation.html.

[3] Nizar Habash and Bonnie Dorr. Large-Scale Language Independent Generation Using Thematic Hierarchies. In *Proceedings of MT Summit VIII, Santiago de Compostella, Spain*, 2001.

[4] Beth Levin. *English Verb Classes and Alternations: A Preliminary Investigation.* University of Chicago Press, Chicago, IL, 1993.

[5] Dekang Lin. Principle-Based Parsing without Overgeneration. In *Proceedings of ACL-93*, pages 112–120, Columbus, Ohio, 1993.

[6] Dekang Lin. Dependency-Based Evaluation of MINIPAR. In *Proceedings of the Workshop on the Evaluation of Parsing Systems, First International Conference on Language Resources and Evaluation*, Granada, Spain, May 1998.

[7] R. Mitten. *Computer-Usable Version of Oxford Advanced Learner's Dictionary of Current English.* Oxford Text Archive, 1992.

[8] P. Procter. *Longman Dictionary of Contemporary English.* Longman, London, 1978.